



組込み機器におけるセキュアOS導入評価

TOSHIBA Corp.

Keijiro Yano

2006/10/27



Contents

- セキュアOSとは？
- Linux用のセキュアOSにはどんなものが？
- 比較的容易に評価出来そうなものは？
- SE Linux vs. LIDS
 - 導入要件
 - セキュリティ機能
 - メモリ使用量
 - 起動時間
- 問題点は？
- 今後の課題



セキュアOSとは？

- アクセス制御機能を強化し、侵入攻撃に対する耐性を高めたOS。root権限を悪用される危険性があるため、
 - MAC(強制アクセス制御)を導入して、すべてのユーザにアクセス制御を実施する。
 - プロセス毎のアクセス制御を導入して、すべての権限のプロセスにアクセス制御が実施される。
 - 権限昇格を制御して、不要な権限を与えないように制御する。

不正アクセスされた場合でも、
被害を最小限に抑えられるようにする。





Linux用セキュアOS

■ SE Linux

- 米国NSA (National Security Agency)が中心に開発。
- LSM (Linux Security Module)を利用した、Linuxカーネル用セキュリティ拡張モジュール。
- Linux 2.6系に標準で組み込まれている。

■ LIDS (Linux Intrusion Detection System)

- Xie Huagang氏、Philippe Biondi氏により1999年10月15日に初版公開。
- LSM (Linux Security Module)を利用した、Linuxカーネル用セキュリティ拡張モジュール。
- Linux Kernel 2.4系 / 2.6系それぞれに対するパッチとして公開。



Linux用セキュアOS

■ AppArmor

- Novellが中心に開発。2006年1月にGPLで公開を開始。
- LSM (Linux Security Module)を利用した、Linuxカーネル用セキュリティ拡張モジュール。
- 現時点ではパッチでの提供。

■ TOMOYO Linux

- NTTデータが中心に開発。2005年11月にGPLで公開を開始。
- LSMを利用せず、Linuxカーネルの各バージョンへのパッチとして提供。

■ Umbrella

- PDAなどのCE機器向けに考案されたセキュリティモジュール。GPLライセンス。
- LSM (Linux Security Module)を利用した、Linuxカーネル用セキュリティ拡張モジュール。

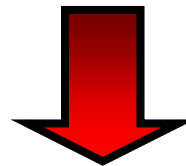
■ LOMAC



SE Linux vs. LIDS

■ 評価対象の選択条件

- Linux Kernel 2.6.10で動作可能なこと。
- パッチ適用が必要でも、容易にパッチがあてられること。
- 使用方法等を示したドキュメントが手に入りやすいこと。



SE Linux vs. LIDS

Kernel 2.6.10に含まれているものを使用

LIDS 2.2.1 を使用(2005/08/30リリース)
最新版は、LIDS 2.2.2



参考文献

- Linuxコンソーシアム セキュリティ部会 成果物
「セキュアOSの評価項目 (Ver.1.0)」

http://www.linuxcons.gr.jp/pdf/sec04_output.pdf



組込み用途を考えた機能比較 (SE Linux vs. LIDS)

- 導入要件
- セキュリティ機能
- メモリ使用量
- 起動時間



組込み用途を考えた機能比較 (SE Linux vs. LIDS)

～導入要件～

	SE Linux	LIDS
Kernel Version	2.6	2.4 / 2.6
CPU Architecture	依存性なし	依存性なし
Filesystem依存性	xattr対応必須	依存性なし
BusyBox拡張	必須	不要
専用ライブラリ	要	不要

拡張属性を制御するためには、専用のライブラリを利用する必要がある。ただし通常は、自製のプログラムにリンクする必要なし。



Filesystemにおける問題点

我々の評価システムではcramfs使用が必須！

■ SE Linuxを使用するためにcramfsへの拡張が必要

- cramfsにはxattr情報を保存する領域が確保されていない。
 - xattr情報を保存できるよう拡張実装。
 - ただし、起動後、cramfsに対するxattr情報の指示必須。

■ LIDSでは、inode番号の扱いに注意が必要

- LIDSはinode番号毎にセキュリティ情報を管理している。
cramfsはファイル・サイズを元にしてinode番号を付与するため、セキュリティ設定ファイルの中身が変わると、inode番号も変わってしまう場合あり。
 - ※ inode番号の割り振り直しをしたいけど、そのたびにセキュリティ設定ファイルのサイズが変わって…という危険がある。



組込み用途を考えた機能比較 (SE Linux vs. LIDS)

～セキュリティ機能～

	SE Linux	LIDS
アクセス制御の粒度	52のオブジェクトクラス、のべ210のアクセスベクタを設定可能。	ファイル/ディレクトリに対して4、プロセスに対して31のアクセス制御設定可能。
ファイル・ディレクトリに対するアクセス制御	○	○
リンクに対するアクセス制御	○	×
スペシャル・ファイルに対するアクセス制御	○	△
パイプに対するアクセス制御	○	△
プロセス間通信に対するアクセス制御	○	△
カーネル・ログに対するアクセス制御	○	△
カーネル・モジュールのロード/アンロード制御	○	○
プロセス毎のアクセス制御	○	△
ユーザ毎のアクセス制御	○	×



組込み機器におけるLIDSの問題点 ～セキュリティ機能～

■ リンクに対するアクセス制御が不可能

LIDSでは、正規化されたファイル名でinode番号を取得し、inode番号毎にセキュリティ設定を保存するため。

BusyBoxが提供する各コマンドはシンボリックリンクで提供される。

→ 個々のコマンドにセキュリティ設定を行うことが出来ない。

■ スペシャル・ファイルに対するアクセス制御が一部不可能

cramfsでは、スペシャル・ファイルに同じinode番号を割り当てるため。

→ 個々にセキュリティ設定を行うことが出来ない。

■ プロセス間通信に対するアクセス制御の機能が不十分

Linux標準で利用できるケーパビリティで設定可能なレベルと、シグナルからの保護しか能力がない。



組み込み用途を考えた機能比較 (SE Linux vs. LIDS) ～起動時間の増加量～

	Boot Time 1	Boot Time 2
SE Linux	6.153 msec	914.642 msec
LIDS	29.555 msec	129.261 msec

■ Boot Time 1

セキュアOSを組み込んでいないLinux Kernelが起動する時間と、
セキュアOSを組み込んだ時にLinux Kernelが起動する時間の差。

■ Boot Time 2

セキュリティ設定情報の読み込み、およびその設定に要する時間。

※本測定用に、両セキュアOSでほぼ同等のセキュリティ設定ファイルを作成し、測定した。



起動時間における問題点

■ Boot Time 1(Kernel起動時間)について

LIDSの計測時間には、セキュリティ設定ファイル(BOOTステート)の読み込み時間が含まれている。セキュリティ設定ファイルの読み込み方が、固定長でファイルを読み込み、

「バッファが足りなければ拡張して再度読み直し」

を繰り返していて非効率。

SE Linux では、初期化処理(LSMへの登録)のみで Boot Time2 の区間(/sbin/init内)でセキュリティ設定の読み込み、設定を行っている。



起動時間における問題点

■ Boot Time 2(セキュリティ設定時間)について

- － セキュリティ設定に大きく依存して変動する。

【SE Linuxのセキュリティ設定】

記述していないものは、基本的にアクセスが制限されることを意味する。

【LIDSのセキュリティ設定】

記述していないものは、基本的にアクセスが可能であることを意味する。

今回の測定では、shell上の基本操作がすべて行えるようなセキュリティ設定を行ったため、SE Linuxのポリシー設定はLIDSと比較して大きくなる。



起動時間における問題点

■ Boot Time 2(セキュリティ設定時間)について

- SE Linux
 - cramfsへのxattr書込時間も含まれているため、起動時間へのインパクト大。本来はファイルシステムイメージに含まれる。
→ 改善のためには、mkcramfsへの改造、inode拡張など、修正量大。
- LIDS
 - POSTBOOTステートへの移行時間が含まれており、再度のセキュリティ設定ファイルの読み込み時間が含まれている。

SE Linux/LIDSのどちらでも、/sbin/initで実行されるシステムコールについて、セキュリティチェックが始まるため、オーバヘッドとなって起動時間に影響する。



組込み用途を考えた機能比較 (SE Linux vs. LIDS) ～メモリ使用量～

■ カーネル・サイズの増加量

	text	data	bss	Total
SE Linux	+101,907	+8,212	+4,096	+114,215
LIDS	+38,960	+12,312	+1,785,856	+1,837,128

単位 : bytes

■ 起動直後の使用メモリ増加量

	使用可能メモリ容量	空きメモリ
SE Linux	- 264 KB	- 1,576 KB
LIDS	- 2,780 KB	- 3,220 KB

メモリ上にファイルシステムイメージを置き、そこにセキュリティ設定ファイルを保持しているため、ほぼ同等のセキュリティ設定を作成して測定した。



組込み機器におけるLIDSの問題点 ～メモリ使用量～

■ カーネル・サイズにおけるbssの増加量が非常に大きい。

LIDSでは、セキュリティ設定情報を、グローバル変数で宣言した大きな配列に保持しているため。

組込み機器では、ちょっと大きすぎると思われるため、下記の改善を実施してみた。

- ・ACLファイル作成時にサイズ情報を追加。
- ・KernelがACLファイルを読み込む際にサイズ情報を参照し、必要な分だけメモリ確保して、ACL情報を保存する。

上記の修正により、

bss増加量は0に

起動時間は 20msec 短縮



組込み用途を考えた機能比較 (SE Linux vs. LIDS) ～LIDS改善後の起動時間～

	Boot Time 1	Boot Time 2
SE Linux	6.153 msec	914.642 msec
LIDS	29.555 msec	129.261 msec
LIDS改	5.983 msec	131.292 msec

■ Boot Time 1

セキュアOSを組み込んでいないLinux Kernelが起動する時間と、セキュアOSを組み込んだ時にLinux Kernelが起動する時間の差。

■ Boot Time 2

セキュリティ設定情報の読み込み、およびその設定に要する時間。

※本測定用に、両セキュアOSでほぼ同等のセキュリティ設定ファイルを作成し、測定した。



組込み用途を考えた機能比較 (SE Linux vs. LIDS) ～LIDS改善後のメモリ使用量～

■ カーネル・サイズの増加量

	text	data	bss	Total
SE Linux	+101,907	+8,212	+4,096	+114,215
LIDS	+38,960	+12,312	+1,785,856	+1,837,128
LIDS改	+42,620	+24,600	0	+67,220

単位 : bytes

■ 起動直後の使用メモリ増加量

	使用可能メモリ容量	空きメモリ
SE Linux	- 264 KB	- 1,576 KB
LIDS	- 2,780 KB	- 3,220 KB
LIDS改	-1,048 KB	- 1,240 KB



まとめ

- 組込み機器においても、ネットワーク接続機能を持った製品が多く開発されてきており、**セキュリティ対策への関心は高まりつつある。**
- 組込み用評価システム上でLinux用の代表的なセキュアOSであるSE Linux／LIDSを動作させ、導入要件、セキュリティ設定機能について実機上で調査し、比較を行った。
- あわせて、評価システム上での起動時間、メモリ使用量の変化量について測定した。



まとめ

■ SE LinuxとLIDSで比較すると、

■ 導入において...

➢ LIDSが優位。SE Linuxでは既存の開発環境に**SE Linux用の対応が必須**。

■ 機能において...

➢ SE Linuxが優位。SE Linuxは**機能が非常に豊富**であり、機能面では十分。
反面、**意図した設定を作成するのが難しい**。

➢ LIDSは設定作成は容易であるが、**機能が不十分である場合あり**。

■ メモリ使用量において...

➢ SE Linuxが優位。LIDSでは導入するだけで3MBを消費する。

■ 性能において...

➢ LIDSが優位。SE Linuxでは、システムコールのオーバーヘッドが大きくなる傾向にある。



今後の課題

■ AppArmorの評価を行いたい。

- Kernel 2.6.10 には簡単にパッチ適用出来ず、今回は断念。
- LIDS や SE Linux とは違って、プログラム単位で セキュアOS 機能を有効にするか、学習機能を有効にするか、セキュアOS 機能を使わないか、が指定できるらしい。
- プログラムファイルのパス名単位に、
保持させる Linux ケーパビリティ
アクセス制御を設定したいファイルのパス名リスト
を指定してアクセス制御をかけるようである。
- セキュリティ設定ファイルの文法は比較的簡単。
- パス名単位での管理は良くない、という意見もあるようである。



今後の課題

- システム設計、アプリケーション開発者と一緒に考えていく必要がある。
OSだけで頑張ってみても、意味のない機能になる可能性がある。
- 検査方法が悩ましい。
 - 侵入評価用のツールだけで検証出来ない。
 - セキュリティ機能の機能検査はどうやって実施する？



CE Linux Forum

CELF Japan Technical Jamboree #11

END